

ℓ_1 Regularized Gradient Temporal-Difference Learning

Dominik Meyer

Hao Shen

Klaus Diepold

DOMINIK.MEYER@TUM.DE

HAO.SHEN@TUM.DE

KLDI@TUM.DE

Institute for Data Processing, Technische Universität München, Germany

Abstract

In this paper, we study the Temporal Difference (TD) learning with linear value function approximation. It is well known that most TD learning algorithms are unstable with linear function approximation and off-policy learning. Recent development of *Gradient TD* (GTD) algorithms has addressed this problem successfully. However, the success of GTD algorithms requires a set of well chosen features, which are not always available. When the number of features is huge, the GTD algorithms might face the problem of overfitting and being computationally expensive. To cope with this difficulty, regularization techniques, in particular ℓ_1 regularization, have attracted significant attentions in developing TD learning algorithms. The present work combines the GTD algorithms with ℓ_1 regularization. We propose a family of ℓ_1 regularized GTD algorithms, which employ the well known soft thresholding operator. We investigate convergence properties of the proposed algorithms, and depict their performance with several numerical experiments.

Keywords: Reinforcement Learning (RL), linear function approximation, Gradient Temporal-Difference (GTD) learning, Iterative Soft Thresholding (IST).

1. Introduction

One fundamental problem in Reinforcement Learning (RL) is to learn the long-term expected reward, i.e. the value function, which can consequently be used for determining a good control policy, cf. [Sutton and Barto \(1998\)](#). In the general setting with large or infinite state space, exact representation of the actual value function is often inhibitive computationally expensive or hardly possible. To overcome this difficulty, function approximation techniques are employed for estimating the value function from sampled trajectories. The quality of the learned policy depends significantly on the chosen function approximation technique.

In this paper, we consider the technique of *linear value function approximation*. The value function is represented or approximated as a linear combination of a set of features, or basis functions. These features are generated from the sampled states via either some heuristic constructions, e.g. [Bradtke and Barto \(1996\)](#); [Keller et al. \(2006\)](#), or kernel-based approaches, e.g. [Taylor and Parr \(2009\)](#). A common approach generates firstly a vast number of features, which is often much larger than the number of available samples, and then chooses automatically relevant features to approximate the actual value function. Unfortunately, such approaches may fail completely due to overfitting. To cope with this situation, regularization techniques are necessarily to be employed. Other than the simple ℓ_2 regu-

larization, which penalizes the smoothness of the learned value function, e.g. Farahmand et al. (2008), in this work we focus on ℓ_1 regularization. The ℓ_1 regularization often produces sparse solutions, thus can serve as a method of automatic feature selection for linear value function approximation.

This work focuses on the development of Temporal Difference (TD) learning algorithms, cf. Bradtke and Barto (1996). Recent active researches on applying ℓ_1 regularization to TD learning have led to a various number of effective algorithms, e.g. Loth et al. (2007); Kolter and Ng (2009); Johns et al. (2010); Geist and Scherrer (2012); Hoffman et al. (2012). It is important to notice that ℓ_1 minimization has been extensively studied in the areas of compressed sensing and image processing, and many efficient ℓ_1 minimization algorithms have been developed, cf. Candés and Romberg (2007); Zibulevsky and Elad (2010). Very recently, two advanced ℓ_1 minimization algorithms have been adapted to the TD learning, i.e. the Dantzig selector based TD algorithm from Geist et al. (2012) and the orthogonal matching pursuit based TD algorithm developed in Painter-Wakefield and Parr (2012a).

On the other hand, most TD learning algorithms are known to be unstable with linear value function approximation and off-policy learning. By observing the fact that most original forms of TD algorithms are not true gradient descent methods, a new class of intrinsic gradient TD (GTD) learning algorithms with linear value function approximation are developed and proven to be stable, cf. Sutton et al. (2008, 2009). However, it is important to know that success of GTD algorithms might be limited due to the fact that the GTD family requires a set of well chosen features. In other words, the GTD algorithms are in potential danger of overfitting. The key contribution of the present work is the development of a family of ℓ_1 regularized GTD algorithms, referred to as *GTD-IST* algorithms. Convergence properties of the proposed algorithms are investigated from the perspective of stochastic optimization.

The paper is outlined as follows. In Section 2, we briefly introduce a general setting of TD learning and provide some preliminaries of TD objective functions. Section 3 presents a framework of ℓ_1 regularized GTD learning algorithms, and investigates their convergence properties. In Section 4, several numerical experiments depict the practical performance of the proposed algorithms, compared with several existing ℓ_1 regularized TD algorithms. Finally, a conclusion is drawn in Section 5.

2. Notations and Preliminaries

In this work, we consider a RL process as a Markov Decision Process (MDP), defined as a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where \mathcal{S} is a set of possible states of the environment, \mathcal{A} is a set of actions of the agent, $P: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ the conditional transition probabilities $P(s, a, s')$ over state transitions from state s to state s' given an action a , $r: \mathcal{S} \rightarrow \mathbb{R}$ is a reward function assigning immediate reward r to a state s , and $\gamma \in [0, 1]$ is a discount factor.

2.1. TD Learning with Linear Function Approximation

The goal of a RL agent is to learn a mapping from states to actions, i.e. a *policy* $\pi: \mathcal{S} \rightarrow \mathcal{A}$, which maximizes the value function $V^\pi: \mathcal{S} \rightarrow \mathbb{R}$ of a state s taking a policy π , defined as

$$V^\pi(s) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \mid s_0 = s, \pi \right]. \quad (1)$$

It is well known that, for a given policy π , the value function V^π fulfills the *Bellman equation*, i.e.

$$V^\pi(s) = r(s) + \gamma \sum_{s'} P(s, \pi(s), s') V^\pi(s'). \quad (2)$$

The right hand side of (2) is often referred to as the *Bellman operator* for policy π , denoted by $\mathcal{T}V^\pi(s)$. In other words, the value function $V^\pi(s)$ is the fixed point of the Bellman operator $\mathcal{T}V^\pi(s)$, i.e. $V^\pi(s) = \mathcal{T}V^\pi(s)$.

When the state space is too large or infinite, exact representation of the value function is often practically unfeasible. Function approximation is thus of great demand for estimating the actual value function. A popular approach is to construct a set of features by the map $\phi: \mathcal{S} \rightarrow \mathbb{R}^k$, which are called the *features* or *basis functions*, and then to approximate the value function by a linear function. Concretely, for a given state s , the value function is approximated by

$$V(s) \approx (\phi(s))^\top \theta =: V_\theta, \quad (3)$$

where $\theta \in \mathbb{R}^k$ is a parameter vector. In the setting of TD learning, the parameter θ is updated at each time step t , i.e. for each state transition and the associated reward (s_t, r_t, s'_t) . Here, we consider the simple one-step TD learning with linear function approximation, i.e. $\lambda = 0$ in the framework of TD(λ) learning. The parameter θ is updated as follows

$$\theta_{t+1} = \theta_t + \alpha_t \delta_t \phi_t, \quad (4)$$

where $\alpha_t > 0$ is a sequence of step-size parameters, and δ_t is the simple TD error

$$\delta_t = r_t + \theta_t^\top (\gamma \phi'_t - \phi_t). \quad (5)$$

Note, that the TD error δ_t can be considered as a function of the parameter θ_t . By abuse of notation, in the rest of the paper we also denote $\delta_\theta = \delta(\theta) := r + \theta^\top (\gamma \phi' - \phi)$.

2.2. Three Objective Functions for TD Learning

In order to find an optimal parameter θ^* via an optimization process, one has to define an appropriate objective function, which accurately measures the correctness of the current value function approximation, i.e. how far the current approximation is away from the actual TD solution. In this subsection, we recall three popular objective functions for TD learning.

Motivated by the fact that the value function is the fixed point of the Bellman operator for a given policy, correctness of an approximation V_θ can be simply measured by the TD error itself, i.e.

$$J_1: \mathbb{R}^k \rightarrow \mathbb{R}, \quad J_1(\theta) := \frac{1}{2} \|V_\theta - \mathcal{T}V_\theta\|_D^2 = \frac{1}{2} (\mathbb{E}[\delta_\theta])^2, \quad (6)$$

where $D \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ is a diagonal matrix, whose components are some state distribution. This cost function is often referred to as the *Mean Squared Bellman Error* (MSBE). Ideally, the minimum of the MSBE function admits a good value function approximation. Unfortunately, it is well known that, in practice, the performance of an approximation V_θ depends

on the pre-selected feature space $\mathcal{H} := \{\Phi\theta | \theta \in \mathbb{R}^k\}$, i.e. the span of the features $\Phi := \phi(\mathcal{S})$. By introducing the projector as

$$\Pi = \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D, \quad (7)$$

the so-called *Mean Squared Projected Bellman Error* (MSPBE) is often preferred

$$\begin{aligned} J_2: \mathbb{R}^k \rightarrow \mathbb{R}, \quad J_2(\theta) &:= \frac{1}{2} \|V_\theta - \Pi\mathcal{T}V_\theta\|_D^2 \\ &= \frac{1}{2} \mathbb{E}[\delta_\theta\phi]^\top \mathbb{E}[\phi\phi^\top]^{-1} \mathbb{E}[\delta_\theta\phi]. \end{aligned} \quad (8)$$

Minimizing the MSPBE function finds a fixed point of the projected Bellman operator in the feature space \mathcal{H} , i.e. $V_\theta = \Pi\mathcal{T}V_\theta$.

Finally, we present a less popular objective function for TD learning. Recall the TD parameter update as defined in (4). The vector $\mathbb{E}[\delta_\theta\phi] \in \mathbb{R}^k$ in the second summand can be considered as an error for a given θ . It is expected to be equal to zero at the TD solution. Hence, one can use the ℓ_2 norm of this vector, defined as

$$J_3: \mathbb{R}^k \rightarrow \mathbb{R}, \quad J_3(\theta) = \frac{1}{2} \mathbb{E}[\delta_\theta\phi]^\top \mathbb{E}[\delta_\theta\phi], \quad (9)$$

as an objective function for TD learning. The function J_3 is referred to as the *Norm of Expected TD Update* (NEU), which is used to derive the original GTD algorithm in [Sutton et al. \(2008\)](#).

3. Stochastic Gradient Algorithms for ℓ_1 Regularized TD Learning

In the first part of this section, we present a general framework of gradient algorithms for minimizing the ℓ_1 regularized TD objective functions. The second subsection develops two ℓ_1 regularized stochastic gradient TD algorithms in the online setting, and investigates their convergence properties from the perspective of stochastic optimization.

3.1. ℓ_1 Regularized TD Learning

Applying an ℓ_1 regularizer to the parameter θ leads to the following objective function

$$F_i(\theta) := J_i(\theta) + \eta \|\theta\|_1, \quad (10)$$

where $i \in \{1, 2, 3\}$ and $\|\theta\|_1 = \sum_i |\theta_i|$ denotes the ℓ_1 norm of a vector $\theta = [\theta_1, \dots, \theta_k]^\top \in \mathbb{R}^k$. Here, the scalar $\eta > 0$ weighs the regularization term $\|\theta\|_1$, and balances the sparsity of θ against the TD objective function J_i . The *iterative soft thresholding* (IST) algorithm is nowadays one classic algorithm for minimizing the cost function (10). It can be interpreted as an extension of the classical gradient algorithm. Due to its high popularity, we skip the derivation of the IST algorithm, and refer to [Zibulevsky and Elad \(2010\)](#) and the references therein for further reading.

Given $x \in \mathbb{R}^m$ and $\nu > 0$, the *soft thresholding operator* applied to x is defined as

$$\begin{aligned} \Psi_\nu(x) &:= \text{sgn}(x) \odot \max\{|x| - \nu, 0\} \\ &= \begin{cases} x - \text{sgn}(x)\nu, & \text{if } |x| > \nu, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \quad (11)$$

where $\text{sgn}(\cdot)$ and $\max(\cdot)$ are entry-wise, and \odot is the entry-wise multiplication. Then, minimization of the objective function (10) can be achieved via applying the soft thresholding operator iteratively. Straightforwardly, we define the IST based TD update as follows

$$\theta_{t+1} = \Psi_{\alpha_t \eta}(\theta_t - \alpha_t \nabla J_i(\theta_t)), \quad (12)$$

where $\alpha_t > 0$, and $\nabla J_i(\theta_t)$ denotes the gradient update of $J_i(\theta_t)$. Specifically, the gradient updates of the three objective functions are given as

$$\begin{cases} \nabla J_1(\theta_t) = \mathbb{E}[\delta_t] \mathbb{E}[(\gamma \phi'_t - \phi_t)], \\ \nabla J_2(\theta_t) = \mathbb{E}[(\gamma \phi'_t - \phi_t) \phi_t^\top] (\mathbb{E}[\phi_t \phi_t^\top])^{-1} \mathbb{E}[\delta_t \phi_t], \\ \nabla J_3(\theta_t) = \mathbb{E}[(\gamma \phi'_t - \phi_t) \phi_t^\top] \mathbb{E}[\delta_t \phi_t]. \end{cases} \quad (13)$$

We refer to this family of algorithms as TD-IST algorithms. Note that IST has been employed in developing fixed point TD algorithms in [Painter-Wakefield and Parr \(2012b\)](#), whereas in this work we focus on developing intrinsic gradient TD algorithm.

3.2. Stochastic GTD-IST Algorithms

The TD-IST algorithms presented in the previous subsection are only applicable in the batch setting. In some real applications, it is certainly favorable to have them working online. Stochastic gradient descent algorithms can be developed straightforwardly to minimize the ℓ_1 regularized TD objective functions.

Now let us consider the online setting, i.e. given a sequence of data samples ϕ_1, ϕ_2, \dots . In the form of stochastic gradient descent, we propose a general form of parameter update as

$$\theta_{t+1} = \Psi_{\alpha_t \eta}(\theta_t - \alpha_t \tilde{\nabla} J_i(\theta_t)), \quad (14)$$

where $\tilde{\nabla} J_i(\theta_t)$ denotes the stochastic gradient updates of $J_i(\theta_t)$, or their appropriate stochastic approximations, cf. [Sutton et al. \(2008, 2009\)](#). To investigate convergence properties of the proposed algorithms requires results from [Duchi and Singer \(2009\)](#), which develops a general framework for analyzing empirical loss minimization with regularizations. We adapt the result in corollary 10 from [Duchi and Singer \(2009\)](#) to our current setting as follows.

Theorem 1 *Let the function $J: \mathbb{R}^k \rightarrow \mathbb{R}$ be smooth and strictly convex and $\theta^* \in \mathbb{R}^k$ be the global minimum of the function $F(\theta) := J(\theta) + \eta \|\theta\|_1$ with $\eta > 0$. If the following three conditions hold: (1) θ^* fulfills $\|\theta_t - \theta^*\|_2 \leq d$ for some constant $d > 0$; (2) $\|\nabla J(\theta_t)\|_2 \leq g$ for some constant $g > 0$; and (3) a stochastic estimate of the gradient $\tilde{\nabla} J(\theta_t)$ fulfills $\mathbb{E}[\tilde{\nabla} J(\theta_t)] = \nabla J(\theta_t)$, then IST based stochastic algorithms converge with probability one to θ^* .*

Let us look at the ℓ_1 regularized NEU function F_3 first. Recall the approximate stochastic gradient update, developed in [Sutton et al. \(2008\)](#), as

$$\tilde{\nabla} J_3(\theta_t) = (\phi_t^\top u_t)(\gamma \phi'_t - \phi_t), \quad (15)$$

with

$$u_{t+1} = u_t + \beta_t(\delta_t \phi_t - u_t), \quad (16)$$

where $\beta_t > 0$ is a step size parameter. We refer to the corresponding algorithm as the *GTD-IST* algorithm. Convergence properties of the GTD-IST algorithm are characterized in the following corollary.

Corollary 2 *If (ϕ_t, r_t, ϕ'_t) is an i.i.d sequence with uniformly bounded second moments, and the matrix $\mathbb{E}[\phi(\gamma\phi' - \phi)^\top] \in \mathbb{R}^{k \times k}$ is invertible, then the GTD-IST algorithm, whose update is specified in (15), converges with probability one to the TD solution.*

Proof Recall the TD error $\delta_\theta = r + \theta^\top(\gamma\phi' - \phi)$. The ℓ_1 regularized NEU cost function F_3 can be written as

$$\begin{aligned} F_3(\theta) &= \mathbb{E}[\delta_\theta \phi]^\top \mathbb{E}[\delta_\theta \phi] + \eta \|\theta\|_1 \\ &= \mathbb{E}[r\phi + \theta^\top(\gamma\phi' - \phi)\phi]^\top \mathbb{E}[r\phi + \theta^\top(\gamma\phi' - \phi)\phi] + \eta \|\theta\|_1. \end{aligned} \quad (17)$$

It is easily seen that the regularized function F_3 is strictly convex if the matrix $\mathbb{E}[\phi(\gamma\phi' - \phi)^\top]$ is invertible. The TD solution is then the global minimum of F_3 . The condition of (ϕ_t, r_t, ϕ'_t) being an i.i.d sequence with uniformly bounded second moments ensures that $\|\nabla J_i(\theta_t)\|_2 \leq g$ holds true for some constant $g > 0$. Finally, applying the fact that the stochastic approximation u_t is a quasi-stationary estimate of the term $\mathbb{E}[\delta\phi]$, cf. [Sutton et al. \(2008\)](#), we have

$$\begin{aligned} \mathbb{E}[\tilde{\nabla} J_3(\theta_t)] &= \mathbb{E}[(\gamma\phi'_t - \phi_t)\phi_t^\top u_t] \\ &= \mathbb{E}[(\gamma\phi'_t - \phi_t)\phi_t^\top] \mathbb{E}[\delta_t \phi_t] \\ &= \nabla J_3(\theta_t). \end{aligned} \quad (18)$$

Then the result follows from Theorem 1. ■

In order to minimize the MSPBE function J_2 , two efficient GTD algorithms are developed in [Sutton et al. \(2009\)](#). Their approximate stochastic updates are defined as

$$\tilde{\nabla} J_2^{(1)}(\theta_t) = (\phi_t^\top w_t)(\gamma\phi'_t - \phi_t), \quad (19a)$$

$$\tilde{\nabla} J_2^{(2)}(\theta_t) = \gamma(\phi_t^\top w_t)\phi'_t - \delta_t \phi_t, \quad (19b)$$

where

$$w_{t+1} = w_t + \beta_t(\delta_t - \phi_t^\top w_t)\phi_t. \quad (20)$$

We refer to the corresponding ℓ_1 regularized GTD algorithms, which employ the updates (19a) and (19b), as GTD2-IST and TDC-IST algorithms, respectively. With no surprises, they share similar convergence properties as the GTD-IST algorithm.

Corollary 3 *If (ϕ_t, r_t, ϕ'_t) is an i.i.d sequence with uniformly bounded second moments, and both $\mathbb{E}[\phi(\phi - \gamma\phi')^\top]$ and $\mathbb{E}[\phi\phi^\top]$ are invertible, then both the GTD2-IST and the TDC-IST algorithms, whose updates are specified in (19), converge with probability one to the TD solution.*

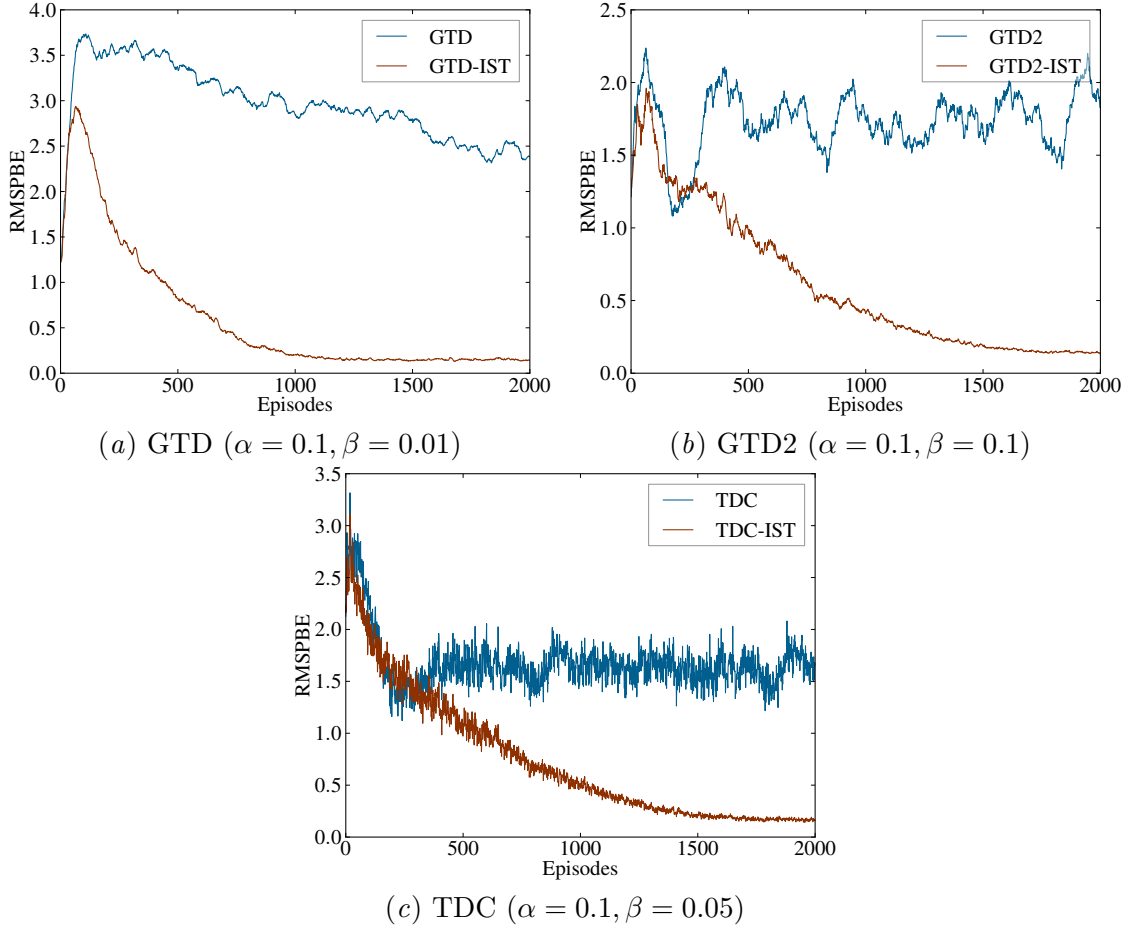


Figure 1: A comparison of IST based GTD learning family ($\eta = 0.001$).

Proof The ℓ_1 regularized MSPBE cost function F_2 can be written as

$$\begin{aligned} F_2(\theta) &= \mathbb{E}[\delta_\theta \phi]^\top \mathbb{E}[\phi \phi^\top]^{-1} \mathbb{E}[\delta_\theta \phi] + \eta \|\theta\|_1 \\ &= \mathbb{E}[r\phi + \theta^\top (\gamma\phi' - \phi)\phi]^\top \mathbb{E}[\phi \phi^\top]^{-1} \mathbb{E}[r\phi + \theta^\top (\gamma\phi' - \phi)\phi] + \eta \|\theta\|_1. \end{aligned} \quad (21)$$

The function F_2 is strictly convex if the matrix

$$\mathbb{E}[\phi(\gamma\phi' - \phi)^\top] \mathbb{E}[\phi \phi^\top]^{-1} \mathbb{E}[(\gamma\phi' - \phi)\phi^\top] \quad (22)$$

is positive definite, i.e. both $\mathbb{E}[\phi(\phi - \gamma\phi')^\top]$ and $\mathbb{E}[\phi \phi^\top]$ are invertible. By the fact that the stochastic approximation w_t is a quasi-stationary estimate of the term $\mathbb{E}[\phi \phi^\top]^{-1} \mathbb{E}[(\gamma\phi' - \phi)\phi^\top]$, cf. [Sutton et al. \(2009\)](#), we get

$$\mathbb{E}[\tilde{\nabla} J_2^{(1)}(\theta_t)] = \mathbb{E}[\tilde{\nabla} J_2^{(2)}(\theta_t)] = \nabla J_2(\theta_t). \quad (23)$$

Then, the result follows straightforwardly from the same arguments as in [Corollary 2](#). \blacksquare

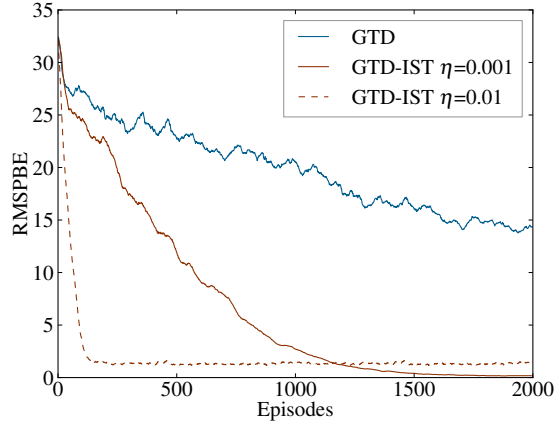


Figure 2: GTD with unfavorable initializations ($\alpha = 0.1, \beta = 0.01$).

4. Numerical Experiments

In this section, we investigate the performance of our proposed ℓ_1 regularized GTD algorithms, compared with two existing ℓ_1 regularized TD algorithms, in both the on-policy and off-policy settings.

4.1. Experiment One: On-Policy Learning

In this experiment, we apply our proposed algorithms to a random walk problem in the chain environment consisting of seven states. There exists only one action and the transition probability of going right or left is equal. A reward of one is only assigned in the rightmost state, which is the terminal state, whereas the rewards are zero everywhere else. The features consist of a binary encoding of the states and ten additional “noisy” features, which are simply Gaussian noise. In this setting, we run three different experiments.

4.1.1. REGULARIZED VS. UN-REGULARIZED

This experiment compares the performance of the proposed ℓ_1 regularized GTD algorithms with their un-regularized counterparts. Figure 1 shows the learning curves of three GTD learning algorithms, namely, GTD, GTD2, and TDC, together with their regularized versions. It is evident that IST based GTD algorithms outperform all their original un-regularized versions respectively. The experimental results demonstrate the effectiveness of IST based GTD learning algorithms.

4.1.2. UNFAVORABLE INITIALIZATIONS

The second experiment investigates the recovery behavior and convergence speed of our proposed algorithms with unfavorable initializations. Here, we only consider the simple GTD-IST algorithm. The parameter vector θ is initialized to have ones for all the noisy features and zeros for all the “good” features. In other words, our experiment starts with the initialization of selecting all the “bad” features. The results in Figure 2 show that the

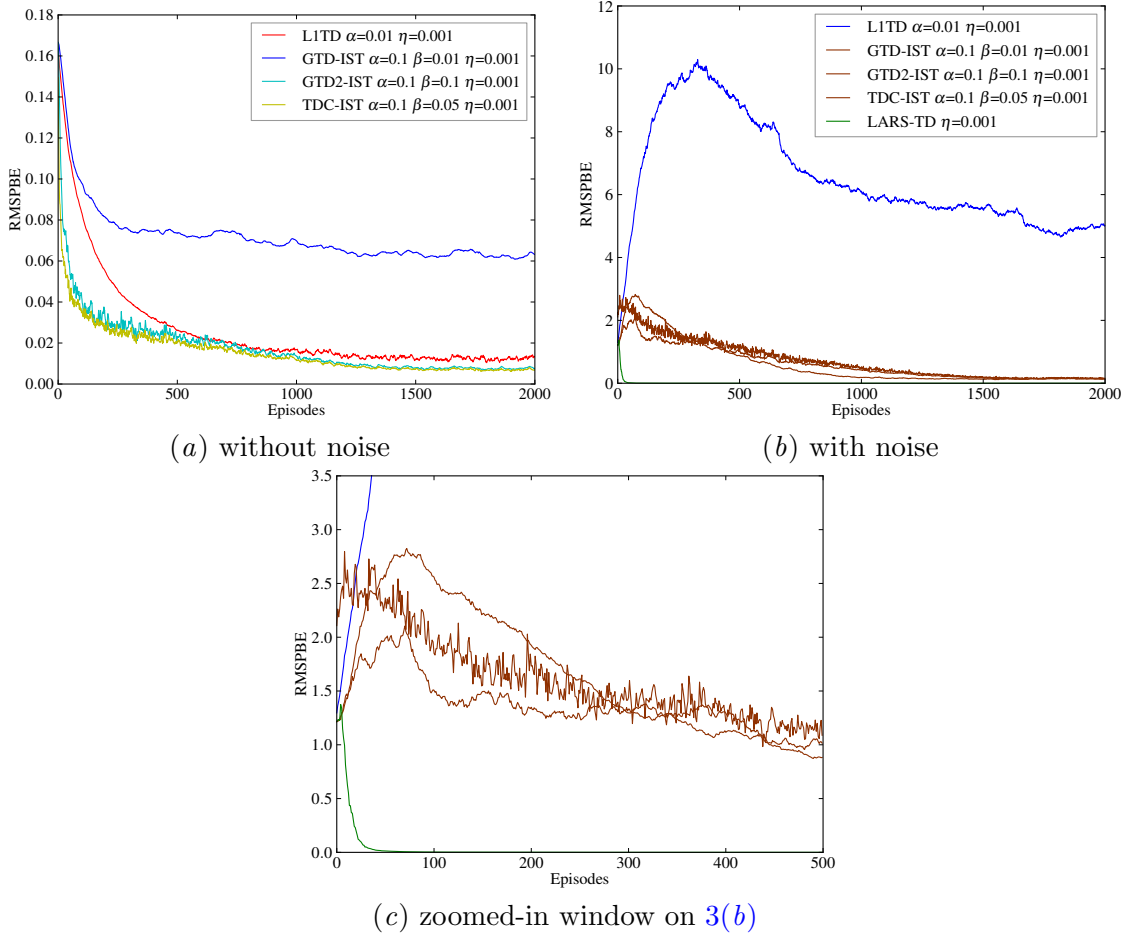


Figure 3: Performance of LARS-TD, L1TD, and the GTD-IST algorithms.

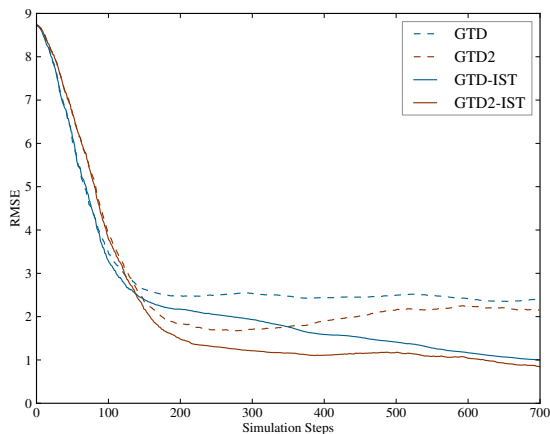
ℓ_1 regularized GTD algorithms, i.e. the GTD-IST algorithm with different parameter value η , converge faster to the correct selection of features than the original GTD algorithm.

4.1.3. GTD-IST ALGORITHMS VS. OTHERS

In the third experiment, we compare the GTD-IST algorithms with the L1TD algorithm from [Painter-Wakefield and Parr \(2012b\)](#) and the LARS-TD algorithm from [Kolter and Ng \(2009\)](#). Results in both Figure 3(a) and 3(b) imply that, with or without noise, all three GTD-IST algorithms outperforms the L1TD algorithm consistently. A closer look at the result in the zoomed-in window in Figure 3(c) shows that the LARS-TD algorithm performs the best with the presence of noise. This might be due to the fact that the LARS-TD algorithm updates, after every 20 episodes, using all the samples available. Nevertheless, without any surprise, a timing experiment shows in Table 1 that the LARS-TD algorithm performs much slower than the other online algorithms.

	L1TD	GTD-IST	GTD2-IST	TDC-IST	LARS-TD
Time (s)	9.2160	9.5565	8.2130	8.4660	118.5490

Table 1: Time measurement of performing 2000 episodes.

Figure 4: Off-policy example ($\alpha = 0.01, \beta = 0.1, \eta = 1$).

4.2. Experiment Two: Off-Policy Learning

To test the performance of the GTD-IST algorithms on the off-policy learning, we employ the well-known star example, proposed in Baird (1995). It consists of seven states with one state being considered as the “center”. In each of the outer states, the agent can choose between two actions: either the “solid” action, which takes it to the center state with probability one, or the “dotted” action, which takes it to any of the other states with equal probability. Reward on all state transitions is equal to zero and the states are represented by tabular features as described in the original setting. We add 20 noisy “Gaussian” features to the state representation. The behavior policy chooses the “solid” action with the probability $1/7$ and the “dotted” otherwise, while the estimation policy chooses always the “dotted” action. The learning curves in Figure 4 shows that both GTD-IST and GTD2-IST algorithms outperform their original counterparts consistently.

5. Conclusions

This work combines the recently developed GTD methods with ℓ_1 regularization, and proposes a family of GTD-IST algorithms. We investigate the convergence properties of the proposed algorithms from the perspective of stochastic optimization. Preliminary experiments demonstrate that the proposed family of GTD-IST algorithms outperform all their original counterparts and two existing ℓ_1 regularized TD algorithms. Being aware of advanced developments in the community of sparse representation, we project to employ further state-of-the-art algorithms of sparse representation to RL. For example, the IST algorithms are usually known to be slow compared to other advanced ℓ_1 minimization algorithms. Applying more efficient ℓ_1 minimization algorithms, such as Beck and Teboulle (2009), to TD learning are of great interests as the future work.

Acknowledgements

This work has been partially supported by the International Graduate School of Science and Engineering (IGSSE), Technische Universität München, Germany. The authors would like to thank Christopher Painter-Wakefield for providing us with the Matlab implementation of the L1TD algorithm.

References

- L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceeding of the 12th International Conference on Machine Learning*, pages 30–37, 1995.
- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):1136–1152, 2009.
- S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Maching Learning*, 22(1-3):33–57, 1996.
- E. J. Candés and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2007.
- J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.
- A. M. Farahmand, M. Ghavamzadeh, C. Szepesvári, and S. Mannor. Regularized policy iteration. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21, pages 441–448. The MIT Press, 2008.
- M. Geist and B. Scherrer. ℓ_1 -penalized projected bellman residual. In S. Sanner and M. Hutter, editors, *Recent Advances in Reinforcement Learning*, volume 7188 of *Lecture Notes in Computer Science*, pages 89–101. Springer Berlin Heidelberg, 2012.
- M. Geist, B. Scherrer, A. Lazaric, and M. Ghavamzadeh. A Dantzig selector approach to temporal difference learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.
- M. W. Hoffman, A. Lazaric, M. Ghavamzadeh, and R. Munos. Regularized least squares temporal difference learning with nested ℓ_2 and ℓ_1 penalization. In S. Sanner and M. Hutter, editors, *Recent Advances in Reinforcement Learning*, volume 7188 of *Lecture Notes in Computer Science*, pages 102–114. Springer Berlin Heidelberg, 2012.
- J. Johns, C. Painter-Wakefield, and R. Parr. Linear complementarity for regularized policy evaluation and improvement. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1009–1017, 2010.
- P. W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML’06)*, pages 449–456, 2006.

- J. Z. Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*, pages 521–528, 2009.
- M. Loth, M. Davy, and P. Preux. Sparse temporal difference learning using lasso. In *Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007.
- C. Painter-Wakefield and R. Parr. Greedy algorithms for sparse reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012a.
- C. Painter-Wakefield and R. Parr. L_1 regularized linear temporal difference learning. Technical report, Department of Computer Science, Duke University, 2012b.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.
- R. S. Sutton, Csaba Szepesvári, and H. R. Maei. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximations. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21, pages 1609–1616. The MIT Press, 2008.
- R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning (ICML’09)*, pages 993–1000, 2009.
- G. Taylor and R. Parr. Kernelized value function approximation for reinforcement learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML’09)*, pages 1017–1024, 2009.
- M. Zibulevsky and M. Elad. $L1$ - $L2$ optimization in signal and image processing. *IEEE Signal Processing Magazine*, 27(3):76–88, 2010.